

TP : Premier projet Git sur Colab

Introduction:

Dans ce travail pratique, nous allons apprendre à utiliser **Git** et **GitHub** directement depuis **Google Colab**.

Git est un outil qui permet de **sauvegarder les versions d'un projet** (comme un code ou un document), tandis que GitHub est une **plateforme en ligne** qui permet de **stocker et partager ces projets**.

Ce TP montre les **étapes de base** pour :

- créer un projet dans Colab,
- l'initialiser avec Git,
- enregistrer les modifications avec un *commit*,
- et enfin envoyer le projet sur un dépôt GitHub.

Étape 1 – Créer un dossier de projet:

Commande :

```
mkdir -p mon_projet  
%cd mon_projet  
!echo "Bonjour Git et GitHub depuis Colab" > texte.txt
```

Résultat :

```
/content/mon_projet/mon_projet
```

Explication :

- `mkdir -p mon_projet` → crée un dossier nommé `mon_projet`.
- `%cd mon_projet` → entre dans le dossier.

- `echo ... > texte.txt` → crée un fichier texte contenant la phrase *"Bonjour Git et GitHub depuis Colab"*.

Étape 2 – Initialiser un dépôt Git :

Commande :

```
\git init -b main
```

Résultat :

```
Initialized empty Git repository in /content/mon__projet/mon__projet/.git/
```

Explication :

- On dit à Git de **créer un nouveau dépôt** dans le dossier.
- L'option `-b main` crée la **branche principale** appelée *main*.

Étape 3 – Configurer ton identité :

Commande :

```
\git config user.name "KhoualdiTaki"  
  
\git config user.email "khoualditaki@example.com"
```

Explication :

- On dit à Git **qui fait les sauvegardes**.
- Chaque commit affichera ton **nom** et ton **e-mail**.

Étape 4 – Ajouter les fichiers et créer un commit :

Commande :

```
\git add.  
  
\git commit -m "Premier commit depuis Colab"
```

Résultat :

```
[main (root-commit) 6c07d4f] Premier commit depuis Colab  
  
1 file changed, 1 insertion(+)  
  
create mode 100644 texte.txt
```

Explication :

- `git add .` → prépare tous les fichiers pour la sauvegarde.
- `git commit -m "..."` → crée une **sauvegarde (commit)** avec le message "Premier commit depuis Colab".
- `6c07d4f` → est l'**identifiant unique** de ce commit.

Étape 5 – Connecter le projet à GitHub :

Commande :

```
from getpass import getpass  
  
import os  
  
token = getpass('Entre ton GitHub token : ')  
  
os.environ['GITHUB_TOKEN'] = token  
  
!git remote add origin  
https://$GITHUB_TOKEN@github.com/KhoualdiTaki/mon_projet_colab.git
```

Résultat :

```
Entre ton GitHub token : .....
```

Explication :

- Tu entres ton **token GitHub** (mot de passe spécial).
- On dit à Git où **envoyer le projet** (sur ton dépôt GitHub).

Étape 6 – Envoyer le projet sur GitHub :

Commande :

```
git push origin main --force
```

Résultat :

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 260 bytes | 260.00 KiB/s, done.

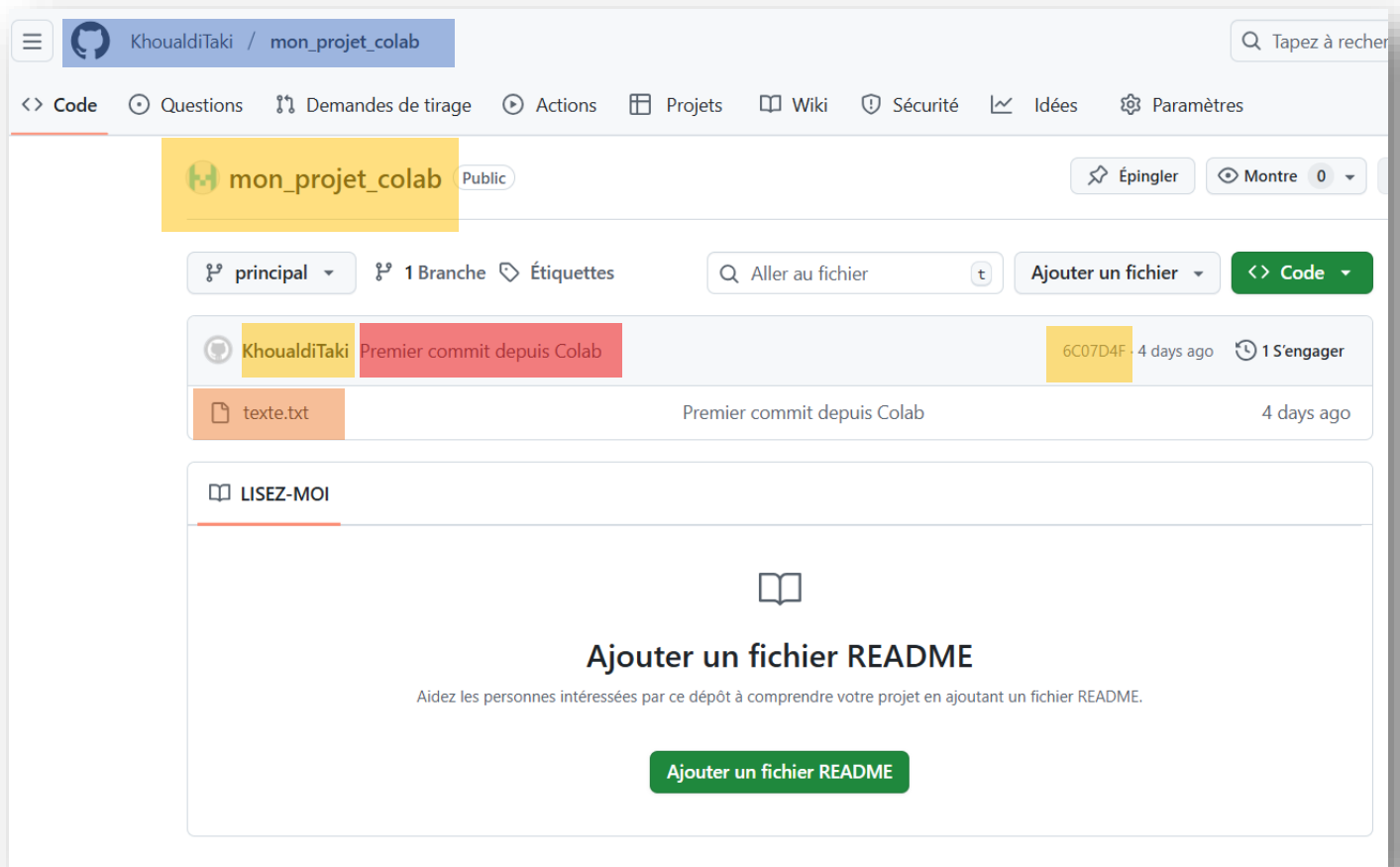
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/KhoualdiTaki/mon__projet__colab.git

+ a4869f7...6c07d4f main -> main (forced update)

Explication :

- `git push origin main` → envoie ton projet sur GitHub dans la branche *main*.
- `--force` → écrase le contenu du dépôt distant si besoin.



Résumé:

À la fin de ce TP :

- Un dossier **mon__projet** a été créé dans Colab.
- Git y a été initialisé.
- Ton premier commit a été fait.
- Le projet a été envoyé sur ton dépôt GitHub :
-> https://github.com/KhoualdiTaki/mon_projet_colab

Conclusion:

Ce TP m'a permis de comprendre comment utiliser **Git** pour suivre les versions d'un projet et **GitHub** pour le stocker en ligne.

Grâce à Google Colab, j'ai pu créer un dépôt, faire un commit et envoyer le projet sur GitHub facilement.

Ces étapes sont les bases du travail collaboratif en programmation.